# Algoritmos y Estructuras de Datos II
# Primer parcial – Miércoles 16 de Noviembre de 2016

**Aclaraciones**

- El parcial es **a libro abierto.**
- Cada ejercicio debe entregarse **en hojas separadas**.
- Incluir en cada hoja el **número de orden asignado, número de hoja, número total de hojas, apellido, nombre y número de Libreta Universitaria.**
- Al entregar el parcial, completar el resto de las columnas en la planilla.
- Cada ejercicio se calificará con **P**romocionado, **A**probado, **R**egular, o **I**nsuficiente.
- El parcial completo está aprobado si el primer ejercicio tiene al menos **A**, y entre los ejercicios 2 y 3 hay al menos una **A**. Para más detalles, ver "Información sobre la cursada" en el sitio Web.

## Ej. 1. Especificación

En un campo de *paintball* compiten en un gran torneo diversos equipos de jugadores. Se desea construir un sistema que mantenga cierta información sobre los equipos que todavía no han sido eliminados. Cada tanto ingresan al predio grupos de integrantes de un mismo equipo (pudiendo haber jugadores del equipo entrante ya presentes en el campo).

Algunos jugadores son más habilidosos que otros. El nivel de habilidad de un jugador se representa con un número natural donde 0 significa "jugador principiante". Dado un equipo, su habilidad total se define como la suma de las habilidades de cada uno de sus jugadores actuales. Durante la batalla sucede que si la cantidad de jugadores de un equipo es inferior a la cantidad de jugadores de otro equipo de habilidad total mayor, entonces el primer equipo es (dolorosamente) eliminado de la batalla de manera inmediata. Dado esto, notar que cuando un grupo de jugadores ingresa al campo, varios equipos pueden resultar eliminados, incluido el equipo del grupo que está ingresando. Por cuestiones comerciales y de balance de la competencia, los organizadores del encuentro prohíben ingresar un grupo de jugadores si eso causara que más de la mitad de la cantidad total de jugadores que había en el campo sea eliminada. Un jugador abandona el campo de juego únicamente cuando su equipo es eliminado: nunca se eliminan jugadores de un equipo individualmente.

El sistema debe poder contestar cuáles son los equipos que compiten actualmente en el campo de juego y cuántos jugadores hay de un equipo determinado.

## Ej. 2. Inducción estructural

Dadas las siguientes funciones sobre árboles binarios ya presentadas en la práctica 2 y el apunte de TADs básicos:

nil? : ab($\alpha$) $\longrightarrow$ bool

$n_1$) nil?(nil) $\equiv$ true

$n_2$) nil?(bin($i, e, d$)) $\equiv$ false

h : ab($\alpha$) $\longrightarrow$ nat

$h_1$) h(nil) $\equiv$ 0

$h_2$) h(bin($i, e, d$)) $\equiv$ máx(h($i$), h($d$)) + 1

#Hojas : ab($\alpha$) $\longrightarrow$ nat

$\#h_1$) #Hojas(nil) $\equiv$ 0

$\#h_2$) #Hojas(bin($i, e, d$)) $\equiv$ **if** nil?($i$) $\wedge$ nil?($d$) **then** 1 **else** #Hojas($i$) + #Hojas($d$) **fi**

#Internos : ab($\alpha$) $\longrightarrow$ nat

$\#i_1$) #Internos(nil) $\equiv$ 0

$\#i_2$) #Internos(bin($i, e, d$)) $\equiv$ **if** nil?($i$) $\wedge$ nil?($d$) **then** 0 **else** #Internos($i$) + #Internos($d$) + 1 **fi**

Se quiere probar por inducción estructural la siguiente propiedad:

$$(\forall a\text{: ab}(\alpha))(\neg\ \text{nil?}(a) \Rightarrow \#\text{Internos}(a) \leq \#\text{Hojas}(a) \times (\text{h}(a) - 1))$$

a) Escribir el predicado unario a utilizar en la demostración.
b) Dar el esquema de inducción a utilizar.
c) Plantear el/los caso/s base/s y resolverlo, justificando cada paso de la demostración.
d) Plantear el/los paso/s inductivo/s, marcando claramente la hipótesis, tesis inductiva y el alcance de los cuantificadores. Resolver justificando cada paso de la demostración.

# Ej. 3. Diseño - Rep & Abs

En la siguiente especificación se ilustra el comportamiento de una carrera de bicicletas. En una carrera participan $n$ ciclistas, numerados de 1 a $n$. Las bicis largan todas juntas y recorren la pista, hasta eventualmente cruzar la meta. Durante la carrera pueden ocurrir choques, en cuyo caso los participantes del choque (que pueden ser cualquier cantidad de bicicletas) finalizan su participación en la carrera. A medida que las bicis cruzan la meta, se les asigna su posición. Todas las bicis accidentadas comparten la posición $n$.

**TAD** CarreraBicicletas

**observadores básicos**

| | | | | |
|---|---|---|---|---|
| participantes | : carreraBicis | $\longrightarrow$ | nat | |
| enCarrera | : carreraBicis $\times$ nat | $\longrightarrow$ | bool | |
| cruzaronMeta | : carreraBicis | $\longrightarrow$ | nat | |
| posicion | : carreraBicis $c$ $\times$ nat $n$ | $\longrightarrow$ | nat | $\{1 \leq n \leq$ participantes$(c)$ $\wedge \neg$ enCarrera$(c,n)\}$ |

**generadores**

| | | | | |
|---|---|---|---|---|
| nuevaCarrera | : nat $n$ | $\longrightarrow$ | carreraBicis | $\{n > 2\}$ |
| cruzarMeta | : carreraBicis $c$ $\times$ nat $n$ | $\longrightarrow$ | carreraBicis | $\{$enCarrera$(c,n)\}$ |
| choque | : carreraBicis $c$ $\times$ conj(nat) $p$ | $\longrightarrow$ | carreraBicis | $\{\forall n, n \in p \Rightarrow_{\text{L}}$ enCarrera$(c,n)\}$ |

**axiomas**

| | | |
|---|---|---|
| participantes(nuevaCarrera($p$)) | $\equiv$ | $p$ |
| participantes(cruzarMeta($c$, $n$)) | $\equiv$ | participantes($c$) |
| participantes(choque($c$, $p$)) | $\equiv$ | participantes($c$) |
| enCarrera(nuevaCarrera($p$), $n$) | $\equiv$ | $1 \leq n \wedge n \leq p$ |
| enCarrera(cruzarMeta($c$,$n'$), $n$) | $\equiv$ | enCarrera($c$,$n$) $\wedge n \neq n'$ |
| enCarrera(choque($c$,$p$), $n$) | $\equiv$ | enCarrera($c$,$n$) $\wedge n \notin p$ |
| cruzaronMeta(nuevaCarrera($p$)) | $\equiv$ | 0 |
| cruzaronMeta(cruzarMeta($c$,$n'$)) | $\equiv$ | 1 + cruzaronMeta($c$) |
| cruzaronMeta(choque($c$,$p$)) | $\equiv$ | cruzaronMeta($c$) |
| posicion(cruzarMeta($c$, $n$), $n'$) | $\equiv$ | **if** $n = n'$ **then** 1 + cruzaronMeta($c$) **else** posicion($c$,$n'$) **fi** |
| posicion(choque($c$,$p$), $n$) | $\equiv$ | **if** $n \in p$ **then** participantes($c$) **else** posicion($c$, $n$) **fi** |

**Fin TAD**

Se decidió utilizar la siguiente estructura para representar el TAD.

carreraBicis **se representa con** estr, donde

estr es tupla ⟨ *posiciones*: conj(tupla ⟨*bici*: nat, *posición*: nat⟩),
*chocadas*: conj(nat)
*enCarrera*: conj(nat) ⟩

donde *posiciones* contiene tuplas indicando qué bicis cruzaron la meta y en qué posición; *chocadas* es el conjunto de bicis que chocaron; y *enCarrera* las bicis que quedan en competencia.

a) Escribir en castellano el invariante de representación.
b) Escribir formalmente el invariante de representación.
c) Escribir formalmente la función de abstracción.

#1

HOJA N°

FECHA

1) TAD paintball

A

| ej 1 | ej 2 | ej 3 |
|---|---|---|
| P | P | - |

Motivos

Ig observacional

$(\forall p_1, p_2 : paintball)\ p_1 =_{obs} p_2 \iff (equipos(p_1) =_{obs} equipos(p_2))$

X Si no puede haber nombres de jugadores repetidos no tiene sentido usar Multiconj, si fuera Multiconj (Habilidad) sí.

TUPLA<NOMBRE, HABILIDAD>

TUPLA<Nombre, Multiconj (Jugador)>

Observadores

equipos : paintball ⟶ Conj(equipos)

~~jugadores : paintball p × equipo e ⟶ multiconj(jugador) {e ∈ equipos(p)}~~

Generadores

crear : ⟶ paintball

ag jugadores : paintball p × multiconj(jugador) mj × nombre m ⟶ paintball

{(no hay equipos(p) ∧ # mj ≠ 0) $\vee_L$

(¬ equipo en juego(p, m) ⇒ # mj ≠ 0 $\wedge_L$

no elimina a más de la mitad (p, mj, m))}

otras op

- nohayequipo : paintball → bool

equipo en juego : paintball p × nombre n → bool

- no elimina a mas de la mitad : paintball p × multiconj(jugador) mj × nombre n → bool

$\{\neg$ nohayequipo(P)$\}$

- nombreseq : paintball p → Conj(nombre)

- habilidad eq : ~~paintball p ×~~ equipo e → nat

$\{\Pi_2(e) \neq \emptyset\}$

esequipodebil : equipo e × Conj(equipo) ce → bool

$\{ce \neq \emptyset \wedge \Pi_2(e) \neq \emptyset \wedge_L e \notin ce\}$

- equipos perdedores : equipo e × Conj(equipo) ce → Conj(equipo)

$\{ce \neq \emptyset \wedge \Pi_2(e) \neq \emptyset\}$

dameNombres : Conj(equipo) → Conj(nombre)

sumar habilidades : multiconj(jugador) m → nat

$\{m \neq \emptyset\}$

- jugadores eq : paintball p × nombre n → multiconj(jugador)

$\{n \in$ nombres eq(P)$\}$

buscar jug : Conj(equipo) ce × nombre n → multiconj(jugador)

$\{n \in$ nombres (ce)$\}$

nombres : Conj(equipo) ce → Conj(nombre)

---

TAD NOMBRE es STRING   TAD HABILIDAD es NAT

TAD JUGADOR es TUPLA<NOMBRE, HABILIDAD>   TAD EQUIPO es TUPLA<NOMBRE, MULTICONJ(JUGADOR)>

#2

HOJA N°

FECHA

- divisa menos : paintball p × Conj(equipo) ce × Multiconj(jugador) mj × nombre n $\longrightarrow$ bool

$\{ \text{equipos}(p) = (ce) \wedge mj \neq \emptyset \}$

- jugtot : paintball p $\longrightarrow$ nat

- esvalidoagregar : equipo e × Conj(equipo) ce × nat n $\longrightarrow$ bool

$\{ e \notin ce \}$

- contar jug : Conj(equipo) $\longrightarrow$ nat

- contar jugadores equipos : Conj(equipo) $\longrightarrow$ nat

Hice 2 veces la misma función sin darme cuenta

#3

HOJA Nº

FECHA

Axiomas

equipos (crear()) $\equiv \emptyset$

equipos (Ag jugadores (P, nj, n)) $\equiv$ if equipos (P) $= \emptyset$

then Ag(<n, nj>, equipos(P)) else

[ if n $\notin$ nombres_eq (P) then [ if es equipodebil(<n,nj>, equipos(P))

then equipos(P) else Ag(<n, nj>, equipos (P)) − equipos perdedores (<n, nj>, equipos (P))

fi ]

else Ag(<n, nj $\cup$ jugadores_eq (P, n)>, equipos (P) − {<n, jugadores_eq(P,n)>})

− equipos perdedores(<n, nj $\cup$ jugadores_eq (P, n)>, equipos(P) − {<n, jugadores_eq(P,n)>})

fi ] fi

~~jugadores(Ag jugadores(p, mj, m))~~

~~jugadores(Ag jugadores(p, mj, m), e) ≡ if $\Pi_1$(e) = m~~

~~then [if e ∉ equipos(p) then mj~~

~~else mj ∪ jugadores(p, e) fi]~~

~~else jugadores(p, e) fi~~

no hay equipos(p) ≡ if #equipos(p) = 0 then true else false fi

equipo en juego(p, n) ≡ if n ∈ nombres eq(p) then true else false fi

nombres eq(p) ≡ dameNombres(equipos(p))

dameNombres(ce) ≡ if ce = ∅ then ∅ else

Ag($\Pi_1$(dameUno(ce)), dameNombres(sinUno(ce))) fi

habilidad eq(e) ≡ sumar habilidades($\Pi_2$(e))

sumar habilidades(m) ≡ if #m = 1 then $\Pi_2$(dameUno(m))

else $\Pi_2$(dameUno(m)) + sumar habilidades(sinUno(m)) fi

#4

$jugadoresq(P, n) \equiv buscarjug(equipo(P), n)$

$buscarjug(ce, n) \equiv$ if $\Pi_1(dameUno(ce)) = n$ then $\Pi_2(dameUno(ce))$ else $buscarjug(sinUno(ce), n)$ fi

como fue el doble conj tiene 1 equipo con ese nombre

$nombres(ce) \equiv$ if $ce = \emptyset$ then $\emptyset$ else $Ag(\Pi_1(dameUno(ce)), nombres(sinUno(ce)))$ fi

$esequipodebil(e, ce) \equiv$ if $\#ce = 1$

then [ if $\#\Pi_2(dameUno(ce)) > \#\Pi_2(e) \land$

$habilidad(dameUno(ce)) > habilidad(e)$

then true else false fi ]

else [ if $\#\Pi_2(dameUno(ce) > \#\Pi_2(e) \land$

$habilidad(dameUno(ce)) > habilidad(e)$

then true

else $esequipodebil(e, sinUno(ce))$ fi ] fi

Era más simple hacer el caso base en $\emptyset$

equiposperdedores$_t$(e, ce) ≡ if #ce = 1 then

[ if #$\Pi_2$(dameUno(ce)) < #$\Pi_2$(e) ∧

habilidad$_{eq}$(dameUno(ce)) < habilidad$_{eq}$(e)

then Ag(dameUno(ce), ∅) else ∅ fi ]

else [ if #$\Pi_2$(dameUno(ce)) < #$\Pi_2$(e) ∧

habilidad$_{eq}$(dameUno(ce)) < habilidad$_{eq}$(e)

then Ag(dameUno(ce), equiposperdedores$_t$(e, sinUno(ce)))

else equiposperdedores$_t$(e, sinUno(ce)) fi ] fi

noeliminaamasdelamitad(P, mj, n) ≡ eliminamenos(P, equipos$_t$(P), mj, n)

eliminamenos(P, CE, mj, n) ≡ if n ∉ nombres$_{eq}$(P) then

[ if esequipodebil$_t$(<n, mj>, equipos$_t$(P)) then true

else esvalidoagregar(<n, mj>, CE, jugtot(P) + #mj) fi ]

else esvalidoagregar(<n, mj ∪ jugadores$_{eq}$(P, n)>, CE − <n, jugadores$_{eq}$(P, n)>,

jugtot(P) + #mj) fi

#5

HOJA N°

FECHA

$jugtot(P) \equiv contarjug(equipos(P))$

$contarjug(CE) \equiv$ if $CE = \emptyset$ then $0$ else
$\#\Pi_2(dameUno(CE)) + contarjug(sinUno(CE))$ fi

$esvalidoagregar(C, CC, n) \equiv$
if $contarjugadoresequipos(equiposperdedores(C, CC)) + contarjugadoresequipos(equiposperdedores(C, CC)) \leq n$
then true else false fi

$contarjugadoresequipos(CC) \equiv$ if $CC = \emptyset$ then $0$
else $\#\Pi_2(dameUno(CE)) + contarjugadoresequipos(sinUno(CE))$ fi

#6

2) $(\forall a : ab(\alpha))\left(\neg nil?(a) \Rightarrow \#Internos(a) \le \#Hojas(a) \times (h_a(a)-1)\right)$

a) $P() \equiv P(a) \equiv \neg nil?(a) \Rightarrow \#Internos(a) \le \#Hojas(a) \times (h_a(a)-1)$

b) $P(nil) \wedge (\forall i,d : ab(\alpha))(P(i) \wedge P(d)) \Rightarrow$

$$(\forall x : \alpha)\, P(bin(i,x,d))$$

c) caso base: $P(nil)$

$$P(nil) \equiv \neg nil?(nil) \Rightarrow \#Internos(nil) \le \#Hojas(nil) \times (h(nil)-1)$$

$$\neg nil?(nil) \Rightarrow \#Internos(nil) \le \#Hojas(nil) \times (h_a(nil)-1)$$

$n_1$

$$\neg true \Rightarrow \#Internos(nil) \le \#Hojas(nil) \times (h_a(nil)-1)$$

Como $\neg true \equiv false$ por axiomas del TAD bool, reemplazo y me queda

$$false \Rightarrow \#Internos(nil) \le \#Hojas(nil) \times (h_a(nil)-1)$$

true

~~d~~ En una implicación lógica, si el valor del antecedente es falso, el valor de verdad de ~~toda~~ la implicación será verdadero, sin importar el valor del consecuente.

Como en este caso el valor del antecedente es falso, toda la implicación será verdadera, por lo que entonces se cumple

$P(nil)$ ~~[illegible]~~

ya que me queda $P(nil) \equiv true$

d) Paso Inductivo

$$\underbrace{(\forall i,d : ab(\alpha))}\,(P(i) \wedge P(d)) \Rightarrow \underbrace{(\forall x : \alpha)}\,P(bin(i,x,d))$$

Como es un $\forall\, i,d$ con suponer $i$ y $d$ cualesquiera ~~y probar~~ (genéricos) y probar que es válida la implicación me alcanza para demostrar que valen para todos los valores alcanzados por el cuantificador

Como es un $\forall x$, con suponer $x$ cualquiera (genérico) y probar que es válido la implicación me alcanza para demostrar que vale para todos los valores alcanzados por el cuantificador

Como es una implicación, si el antecedente es falso entonces toda la implicación es verdadera, por lo que supongo verdadero el antecedente y busco probar que en este caso es válido el consecuente y así poder demostrar toda la implicación

#7

Entonces quiero ver que es válido $P(bin(i,x,d))$

$$P(bin(i,x,d)) \equiv \neg nil?(bin(i,x,d)) \Rightarrow$$

$$\#Internos(bin(i,x,d)) \le \#Hojas(bin(i,x,d)) \times (h_a(bin(i,x,d))-1)$$

Desarrollo el antecedente

$$\neg nil?(bin(i,x,d))$$

$$\xrightarrow{n_2} \neg false$$

$$\to true$$

Como el antecedente es verdadero *siempre* entonces debo probar que el consecuente también lo es

$$\#Internos(bin(i,x,d)) \le \#Hojas(bin(i,x,d)) \times (h_a(bin(i,x,d))-1)$$

$\#i_2$

$$\text{if } nil?(i) \wedge nil(d) \text{ then } 0 \text{ else } \#Internos(i) + \#Internos(d) + 1 \text{ fi}$$

$\# h_2$

$$\le (\text{if } nil?(i) \wedge nil?(d) \text{ then } 1 \text{ else } \#Hojas(i) + \#Hojas(d) \text{ fi}) \times (h(bin(i,x,d))-1)$$

reescribo la desigualdad

$$\text{if } nil?(i) \wedge nil?(d) \text{ then } 0 \text{ else } \#Internos(i) + \#Internos(d) + 1 \text{ fi}$$

$$\leq \left(\text{if } nil?(i) \wedge nil?(d) \text{ then } 1 \text{ else } \#Hojas(i) + \#Hojas(d) \text{ fi}\right)$$

$$\times \left(h(bin(i,x,d)) - 1\right)$$

$$\Big\downarrow \text{ por } h_2$$

$$\text{if } nil?(i) \wedge nil?(d) \text{ then } 0 \text{ else } \#Internos(i) + \#Internos(d) + 1 \text{ fi}$$

$$\leq$$

$$\left(\text{if } nil?(i) \wedge nil?(d) \text{ then } 1 \text{ else } \#Hojas(i) + \#Hojas(d) \text{ fi}\right) \times \left(max(h(i), h(d)) + 1 - 1\right)$$

$$\Downarrow$$

$$\text{if } nil?(i) \wedge nil?(d) \text{ then } 0 \text{ else } \#Internos(i) + \#Internos(d) + 1 \text{ fi}$$

$$\leq$$

$$\left(\text{if } nil?(i) \wedge nil?(d) \text{ then } 1 \text{ else } \#Hojas(i) + \#Hojas(d) \text{ fi}\right) \times max(h(i), h(d))$$

Ahora tengo que probar 2 casos posibles

$$nil?(i) \wedge nil?(d) \quad \text{y} \quad \neg(nil?(i) \wedge nil?(d))$$

(1) (2)

Pruebo (1), Reemplazando el desarrollo anterior por el caso de (1)

Queda

$$0 \leq 1 \times max(h(i), h(d))$$

#8

Como supuse nil?(i) ^ nil?(d) entonces por $h_1$ vale que

$h(i) \equiv 0$ y $h(d) \equiv 0$ (ya que, i y d son nil)

entonces me queda que $0 \leq 1 \times max(0,0)$

$0 \leq 1 \times 0$ El máximo entre 0 y 0 es 0

$0 \leq 0$ ✓

Llegué a un resultado válido, por lo que vale (1)

Prueba (2), Reemplazando el desarrollo ~~entero~~ por el caso de (2) queda

$\#Internos(i) + \#Internos(d) + 1 \leq (\#Hojas(i) + \#Hojas(d)) \times max(h(i), h(d))$

Ahora me quedan 3 casos

(a) nil?(i) ^ ¬nil(d)

(b) ¬nil(i) ^ nil(d)

(c) ¬nil(i) ^ ¬nil(d)

Prueba (2)(a)

Como vale nil?(i) entonces por $\#i_1$ vale que $\#Internos(i) \equiv 0$

por $\#h_1$ vale que $\#Hojas(i) \equiv 0$ y por $h_1$ vale que $h(i) \equiv 0$

Reemplazando estos 3 valores me queda lo siguiente

~~HI~~ $0 + \#Interna(d) + 1 \leq (0 + \#Hojas(d)) \times max(0, h(d))$

$$\#Interna(d) + 1 \leq \#Hojas(d) \times max(0, h(d))$$

Como 0 es el mínimo nat. y como h(d) devuelve un nat en el peor de los casos $h(d) \equiv 0$, mientras que en los otros $h(d) > 0$, por lo que entonces puedo decir que

$max(0, h(d)) \equiv h(d)$ ya que si $h(d) > 0$ $max(0, h(d)) \equiv h(d)$

y si $h(d) \equiv 0$ $max(0,0) \equiv 0 \equiv h(d)$

$$\#Interna(d) + 1 \leq \#Hojas(d) \times h(d)$$

resto 1 de ambos lados

$$\#Interna(d) \leq \#Hojas(d) \times h(d) - 1$$

Como $\neg nil?(d)$, por HI

$$\#Interna(d) \leq \#Hojas(d) \times (h(d) - 1) \equiv \#Hojas(d) \times h(d) - \#Hojas(d)$$

y como $\#Hojas(d) \times h(d) - \#Hojas(d) \leq$

$$\#Hojas(d) \times h(d) - 1$$

ya que por lema $\#Hojas(d) \geq 1$ cuando $\neg nil?(d)$ en el peor de los casos $\#Hojas(d) \equiv 1$, y en cualquier otro caso el resultado es menor a $\#Hojas(d) \times h(d) - 1$ por lo que puedo acotarlo por ese valor.

Se cumple la implicación para el caso (a)

HOJA N° (5)

FECHA

#9 (b) vale nil(d), por #i1 $\#Interno(d) \equiv 0$, por #h1 $\#Hoja(d) \equiv 0$, por h1 $h(d) \equiv 0$

Reemplazando

~~[illegible]~~

$$\#Interno(i) + 0 + 1 \leq (\#Hoja(i) + 0) \times max(h(i), 0)$$

$max(h(i), 0) \equiv h(i)$ ~~por la regla~~ como en el caso anterior

$$\#Interno(i) \leq \#Hoja(i) \times \cancel{max(h(i),0)}\ h(i) - 1$$

por HI

distribuyendo

$$\#Interno(i) \leq \#Hoja(i) \times (h(i) - 1) \equiv \#Hoja(i) \times h(i) - \#Hoja(i)$$

Y como $\#Hoja(i) \times h(i) - \#Hoja(i) \leq$

$$\#Hoja(i) \times h(i) - 1$$ por el lema y justificando como en el caso anterior

Se llega a un resultado válido por lo que probé (b)

(c)

$\#Internos(i) + \#Internos(d) + 1 \leq (\#Hojas(i) + \#Hojas(d)) \times max(h(i), h(d))$

pasorestando el 1

$\#Internos(i) + \#Internos(d) \leq (\#Hojas(i) + \#Hojas(d)) \times max(h(i), h(d)) - 1$

por HI como $\neg nil?(i)$ y $\neg nil?(d)$

$\#Internos(i) \leq \#Hojas(i) \times (h(i) - 1) \equiv \#Hojas(i) \times h(i) - \#Hojas(i)$

Sumo $\#Int(d)$ en ambos lados de la desig

$\#Internos(i) + \#Internos(d) \leq \#Hojas(i) \times h(i) - \#Hojas(i) + \#Internos(d)$

Por HI

$\#Hojas(i) \times h(i) - \#Hojas(i) + \#Internos(d) \leq$

$\#Hojas(i) \times h(i) - \#Hojas(i) + \#Hojas(d) \times (h(d) - 1) \equiv$

$\equiv \#Hojas(i) \times h(i) - \#Hojas(i) + \#Hojas(d) \times h(d) - \#Hojas(d) \equiv$

$\equiv \#Hojas(i) \times h(i) + \#Hojas(d) \times h(d) - (\#Hojas(i) + \#Hojas(d)) \leq$

como $h(i) \leq max(h(i), h(d))$ y
$h(d) \leq max(h(i), h(d))$

Acoto

$\leq \#Hojas(i) \times max(h(i), h(d)) + \#Hojas(d) \times max(h(i), h(d))$
$- (\#Hojas(i) + \#Hojas(d)) \equiv$

#10

Sacando factor común queda

$$\Xi\left(\#Hojas(i)+\#Hojas(d)\right)\times max(h(i), h(d)) - (\#Hojas(i)+\#Hojas(d))$$

$$\leq (\#Hojas(i)+\#Hojas(d))\times max(h(i), h(d)) - 1$$

ya que por lema

$\neg nil?(i) \Rightarrow \#Hojas(i) \geq 1$

$\neg nil?(d) \Rightarrow \#Hojas(d) \geq 1$

$\Rightarrow \#Hojas(i)+\#Hojas(d) \geq 2 \geq 1$

~~mult. por~~

por lo que como estoy restando la cota por el mínimo valor que pueden tomar que en fact es más ~~chico~~ grande que 1 y por eso luego la acoto por 1

Por lo tanto vale (C)

Como valen (1) y (2)(a)(b) y (c) entonces llegué a un resultado válido en todos los casos por lo que ~~~~ se cumple el paso inductivo y la propiedad queda demostrada.

Lema

$$(\forall a : ab(\alpha))(\neg nil?(a) \Rightarrow \#Hojas(a) \geq 1)$$

<u>PU</u>

$$P(a) \equiv \neg nil?(a) \Rightarrow \#Hojas(a) \geq 1$$

esquema

$$P(nil) \wedge (\forall i,d : ab(\alpha))(P(i) \wedge P(d)) \Rightarrow (\forall x : \alpha)\, P(bin(i,x,d))$$

prueba

CB $P(nil) \equiv \neg nil?(nil) \Rightarrow \#Hojas(nil) \geq 1$

$\overset{n_1}{\equiv} \neg true \Rightarrow \#Hojas(nil) \geq 1$

$\equiv false \Rightarrow \#Hojas(nil) \geq 1$

Como es un implica y el antecedente es falso entonces prueba ~~la implicación~~ el caso base ya que la implicación es verdadera

paso inductivo

$$(\forall i,d : ab(\alpha))(P(i) \wedge P(d)) \Rightarrow (\forall x : \alpha)\, P(bin(i,x,d))$$

Como es un $\forall$, supongo $i,d$ y $x$ cualesquiera

QVQ $P(i) \wedge P(d) \Rightarrow P(bin(i,x,d))$

asumo antecedente verdadero y pruebo que el consecuente es verdadero

#11

$P(bin(i,r,d)) \equiv \neg nil?(bin(i,r,d)) \Rightarrow \#Hojas(bin(i,r,d)) \geq 1$

$\neg false \Rightarrow \#Hojas(bin(i,r,d)) \geq 1$

$true \Rightarrow \#Hojas(bin(i,r,d)) \geq 1$

Como el antecedente es verdadero para que sea verdadera la implicación, el consecuente también lo debe ser

entonces pruebo que

$\#Hojas(bin(i,r,d)) \geq 1$

$\#h_2$

if $nil?(i) \wedge nil?(d)$ then 1 else $\#Hojas(i) + \#Hojas(d) \geq 1$

2 casos

(1) $nil?(i) \wedge nil?(d)$

reemplazo y me queda $1 \geq 1$ ✓ lo cual es verdadero, vale (1)

(2) $\neg(nil?(i) \wedge nil?(d))$ 3 casos

(a) $nil?(i) \wedge \neg nil?(d)$

reemplazo

Como $nil?(i)$ entonces por $\#h_1$

$\#Hojas(i) + \#Hojas(d) \geq 1$

$0 + \#Hojas(d) \geq 1$

$\#Hojas(d) \geq 1$

Como $\neg nil?(d)$ entonces por HI

esto es válido, verdadero

vale (2)(a)

(b) $\neg nil?(i) \wedge nil?(d)$ reemplazo

$$\#Hojas(i) + \#Hojas(d) \geq 1$$

Como nil?(d) por #h$_1$

$$\#Hojas(i) + 0 \geq 1$$

$$\#Hojas(i) \geq 1$$

Como $\neg nil?(i)$ por HI vale esto por lo que vale (b)

(c) $\neg nil?(i) \wedge \neg nil?(d)$

reemplazo $\#Hojas(i) + \#Hojas(d) \geq 1$

Por HI como $\neg nil?(i)$ vale que

$$\#Hojas(i) \geq 1$$

sumo #Hojas(d) en ambos lados del $\geq$

$$\#Hojas(i) + \#Hojas(d) \geq 1 + \#Hojas(d)$$

y $1 + \#Hojas(d) \geq 1 + 1 = 2 \geq 1$

ya que como $\neg nil?(d)$ vale la HI

Por lo tanto llego aquí vale que

$$\#Hojas(i) + \#Hojas(d) \geq 1$$

entonces vale (c)

Como valen (1) y (2) (a) (b) y (c) entonces valen ~~los~~ todos los casos y por lo tanto se cumple el paso inductivo

Por ende vale el lema (ya que valen CB y PI)

NOTA